

HARDWARE/SOFTWARE IMPLEMENTATION OF A PWM WITH ENHANCED
FEATURES USING A STANDARD MICROPROCESSOR

5

The present invention relates to pulse-width-modulation (PWM) waveform generators. More particularly, the present invention relates to an intelligent, programmable PWM generator with enhanced features implemented using standard modules in a microprocessor. The PWM of the present invention is able to handle both variable on/off-timing control and multiple-event interrupts.

A PWM control signal is widely used in power converters with a PWM control to turn on and off power switches. Examples of PWM generators include: a hardware-implemented, duty-cycle-modulated PWM, as illustrated in FIG. 1; a current control circuit with critical-boundary modulation, as illustrated in FIG. 2; and, a software-controlled PWM generator with programmable on and off periods, as illustrated in FIG. 5.

A simple PWM generator circuit with a variable pulse-width and fixed-period control is illustrated in FIG. 1. The control-signal waveforms are illustrated in FIG 2. The inputs to the circuit consist of one saw-toothed signal, V_{saw} , and a reference voltage, V_{err} , which is a fixed voltage for an open-loop circuit and an error voltage from the output of a compensation network for a closed-loop circuit. The PWM signal becomes low when the saw-toothed voltage reaches V_{err} as illustrated in FIG. 2. The pulse-width of the PWM is controlled by V_{err} .

A PWM generator of critical boundary control with peak-current-detect off and zero-current-detect on is illustrated in FIG. 3. The control waveforms are illustrated in FIG. 4. The PWM becomes low when a peak current is detected and the PWM becomes high again when a zero current is detected (ZCD). The PWM can have a variable pulse-width and period.

In addition to PWM generators with hardware implementation in conventional analog circuits, a commonly used PWM generator available in many microprocessors is illustrated in FIG. 5. The microprocessor calculates the switching period and on-time of the PWM signal and pre-loads the counter registers before the next period. The output of a PWM generator is illustrated in FIG. 6. However, in the implementation illustrated, the software cannot handle the external on or off signal as a regular control signal (e.g. in each period continuously).

Many applications require combined features from known PWM generators. For example, in existing peak current mode controlled circuits, the PWM can be forced off by a peak current detection, forced-on by a zero current detection while at the same time a maximum on time, minimum off time, and maximum off time must be satisfied.

Another example of an application requiring combined features of known PWM generators is a digital-controlled circuit that requires two-mode operations depending on load conditions -- e.g. current mode control under a heavy load and voltage mode control under a light load. In a current mode control, the on and off is controlled by external forced on and forced off, in voltage mode control, the on and off is calculated by a software in a microprocessor with a programmable on and off timing.

Another example is a power factor-correction circuit with programmable on-time and zero detection off-time control.

FIG. 7 illustrates a block diagram of a standard timer. Two parameters are preloaded in the timer: period and compare value. The timer automatically reloads timing parameters when the previous counted period is reached or a reload command is sent from the software. The timer starts counting when the counter is preloaded and the enable signal is high. An interrupt signal can be generated when the compared value is reached or the period value is reached. FIG. 8 illustrates the output waveform of the standard timer illustrated in FIG. 7.

None of the prior art PWM devices provides this combination of features in a digitally-controlled circuit using a microprocessor.

Thus, a PWM generator is needed with enhanced features that are implemented in a microprocessor with a software and having standard hardware peripherals. The PWM functions of the present invention are implemented by a software in a microprocessor that handles not only on or off events controlled by external pins, as in FIGs. 1 and 3, but is also programmable on and off timing as illustrated in FIG. 5.

A block diagram of a generic PWM generator according to a preferred embodiment of the present invention has on/off time control and multiple-event on/off actions as illustrated in FIG. 9. In this preferred embodiment, the PWM function with enhanced features uses standard modules in a microprocessor and is implemented by a hardware/software interaction in a microprocessor. No extra hardware is needed in the implementation. The number of force-on and force-off events depends on application requirements and the number of pins can be more than the two pins that are illustrated in FIG. 9.

FIG. 1 illustrates a simple variable pulse-width and fixed-period control circuit.

FIG. 2 illustrates control waveforms with a circuit illustrated in FIG. 1.

FIG. 3 illustrates a control circuit with peak current off and zero current on.

FIG. 4 illustrates control-signal waveforms with the circuit illustrated in FIG 3.

FIG. 5 illustrates a block diagram of a commonly-used PWM generator in MCU.

FIG. 6 illustrates the PWM output waveform of the generator illustrated in FIG. 5.

FIG. 7 illustrates a block diagram of a standard timer.

FIG. 8 illustrates the output of a standard timer.

FIG. 9 illustrates a block diagram of a generic PWM generator with on and off time controls and multiple-event interrupts.

FIG. 10 illustrates a block diagram of a PWM generator using standard units having enhanced features, according to an embodiment of the present invention.

FIG. 11 illustrates a flow chart of multiple-event interrupts for a PWM generator.

An implementation block diagram of a PWM according to a preferred embodiment of the present invention is illustrated in FIG. 10. As illustrated, the PWM generator of the present invention has two parts: a programmable on-and-off-control part 102 and a multiple-event-interrupts part 103. The programmable on-off-control part 102 is implemented by a first and second standard timer in a microprocessor. The multiple-event-interrupts part 103 is implemented by multiple interrupt routines in the software. The programmable PWM generator is controlled by hardware/software interactions.

In a preferred embodiment of the present invention, the two standard timers that are used in the programmable on-off-control part 102 are used for an on-time control 100 and an off-time control 101. These two controls function alternatively. The on-time control 100 is enabled when the PWM is high and the off-time control 101 is enabled when the PWM is low. The on-time control 100 sets the PWM to low when the min_on time is exceeded and either a force_off or a max_on event is triggered. The off-time control sets the PWM to high when the min_off time is exceeded and either a force_on or a max_off event is triggered. The timing parameters are pre-loaded in the inactive control when the other control is active, so there is no delay involved in this loading process.

FIG. 9 is a block diagram of a generic PWM according to the present invention. Illustrated are four registers and two events that can be externally or programmatically controlled (Force_on and Force_off). A pre-determined protocol for a PWM is the on-off pulse and period width(s) and the order of producing these on/off pulses, e.g., a specification for a series of pulses having a given width and period. For example, the period 90 is determined by the value of the Max_on register and the Max_off register or (Max_on register + Max_off register). The on-width 92 is determined by the value of the Max_on register. The off width 93 is determined by the value of the Max_off register. In a given protocol, Force_on and Force_off events can override the register values to a certain extent within a minimum on and maximum on, and minimum off and maximum off period, as illustrated in FIG. 11.

FIG. 11 illustrates a flow chart 120 of a preferred embodiment of the multiple-event-interrupts part 103 that implements a pre-determined protocol for a series of on/off pulses that is produced by a PWM generator. The interrupts associated with this pre-determined protocol are implemented by interrupt-handling software routines resident in a microprocessor. In the preferred embodiment of the programmable on/off control 118, illustrated in FIG. 11, four timer registers are provided 118 two for the timer 1 and the timer 2. Six events can be triggered 119: four by the programmable on/off control software and two externally. Keeping in mind that the timers are active alternately -- meaning that the performance of a. and b. below alternates -- in a preferred embodiment, the flow of processing by the multiple-event interrupts processor 103 is as follows:

a. If either a force_off action occurs (int5 is triggered) or timer1 has been on longer than Tmax_on (int2 is triggered) then int1 is tested at step 111 until it is triggered by the compare value reaching Tmin_on. Then, at step 112 the PWM is set to low or off. Following step 112, at step 113 timer2 is turned on, timer 1 is turned off, int6=force_on is enabled, and int5=force_off is disabled.

b. If either a force_on action occurs (int6 is triggered) or timer2 has been off longer than Tmax_off (int4 is triggered), then int3 is tested at step 115 until it is triggered by the compare value reaching Tmin_off. Then, at step 116 the PWM is set to high or turned on. Following step 116, at step 117 timer1 is turned on, timer 2 is turned off, int5=force_off is enabled, and int6=force_on is disabled.

When some of the events are not required by an application, those events can be disabled easily in the software.

The PWM implementation of a preferred embodiment of the present invention employs standard peripherals of a microprocessor without requiring any special units or extra hardware thus providing maximum flexibility as a PWM generator.

The PWM of a preferred embodiment has the following features:

- Programmable min-on control
- Programmable max-on control
- Programmable min-off control
- Programmable max-off control
- Multiple force-off events control, programmable or external
- Multiple force-on events control, programmable or external
- On/off timing can involve all or some of the parameters or events above.
- All events can be triggered in every PWM period continuously
- No delay time for pre-loading timer registers

While the invention has been explained with reference to specific embodiments of the invention, the explanation is presented by way of illustration only and not in any limiting sense. The invention scope is not restricted to the software and hardware structure described above but includes various modifications and variations that would be apparent to one skilled in the art. Such variations are not to be regarded as a departure from the spirit and scope of the invention, and all such modifications are intended to be included within the scope of the following claims.